Landsat 7
Processing System (LPS)
Interface Definitions Document

November 3, 1995

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

Landsat 7
Processing System (LPS)
Interface Definitions Document

November 3, 1995

Prepared by:

David Sames Date
Software Engineer
Landsat 7 Processing System
Code 514.1
Goddard Space Flight Center

Approved by:

J. B. Martin Date
Ground System Manager
Landsat 7 System
Code 502
Goddard Space Flight Center

Concurred by:

Joy Henegar Date
Project Manager
Landsat 7 Processing System
Code 514.a
Goddard Space Flight Center

R. Obenschain Date
Project Manager
Landsat 7 System
Code 430
Goddard Space Flight Center

TBD Date
Chairman, Code 510 CCB
Goddard Space Flight Center

Abstract

The Landsat 7 Processing System (LPS) provides Landsat 7 data receipt and processing support to the Landsat 7 program, in conjunction with the Earth Science Mission Operations (ESMO) Project. The LPS receives raw wideband data from the Landsat 7 Ground Station, located at the EROS Data Center (EDC), processes it into Level 0R, browse and metadata files, and provides them to the Landsat Processes Distributed Active Archive Center (LP DAAC), also located at the EDC. The interface definitions presented in this document are based on the information contained in the LPS Functional and Performance Specification (F&PS), the LPS System Design Specification, the LPS Operations Concept document, the LPS Software Requirement Specification, and the LPS Preliminary Design Specification.

Keywords:

Landsat 7
Landsat 7 Processing System (LPS)
Landsat 7 Ground Station (LGS)
Landsat Processes Distributed Active Archive Center
(LP DAAC)
Functional and Performance Specification (F&PS)
Mission Operations and Data Systems Directorate
(MO&DSD)
Systems Management Policy (SMP)
Mission Operations & System Development Division
(MO&SD)

Preface

This document defines the interfaces between LPS subsystems. These definitions are based on an analysis of the requirements contained in the LPS Functional and Performance Specification (F&PS), the LPS System Design Specification, the LPS Operations Concept document, and the LPS Software Requirement Specification. This document, once baselined, will be controlled by the MO&SD (Code 510) configuration control board (CCB) and maintained and updated, as required, by the LPS Project.

This interface definitions document was prepared by:

Landsat 7 Processing System Project
Code 510
Goddard Space Flight Center
Greenbelt, MD 20771

GSFC
T. Grubb
J. Henegar
J. Hosler
E. Lee
K. Michael
D. Sames
R. Schweiss
M. Dowdy
G. Toth
J. Rosenberg

SEAS
D. Alban
T. Aslam
B. Boyce
D. Crehan
J. Freeman
S. Gottsagen
M. Huang
J. Lauderdale
C. Liu
N. Patel
R. Shea
R. Tingley
M. Wong
I. Wu

Table of Contents

8 LPS Data Transfer Subsystem

1 Introduction

1.1 Purpose

This Interface Definitions Document (IDD) describes the interfaces among LPS subsystems. The description of each interface includes the purpose of the interface, the interactions between the subsystems, the dataflow structures, and the IPC mechanism(s) used to carry out the interactions.

1.2 Scope

This document describes the interfaces between the subsystems composing the LPS. The scope of this document is a description of the data products, information, and control being transferred across the subsystem interfaces and is based on the LPS Software Requirement Specifications (SRS) and the LPS Preliminary Design Specification completed by the LPS project.

This document is part of the LPS project baseline. It takes effect upon approval by the MO&SD Division (Code 510) LPS Project Configuration Control Board (CCB). Proposed changes to this document require the same level of approval.

1.3 Document Organization

This document is organized as follows:

- Section 1 describes the purpose, scope, and organization of this document; lists the documents referenced; and presents an overview of the LPS subsystem interfaces.

- Section 2 through section 8 describe the various subsystem interfaces. Each section contains an overview of the interface purpose, a description of interactions between the subsystems, a description of its dataflows and contents, and a description of the IPC mechanism(s) used to transfer information across the interface. Each line of the dataflow structure is an item of the data dictionary. The item's decomposition is shown as the next level of indentation below it. Each level of composition is indented from the previous level.

1.4 Applicable Documents

This section lists those documents referenced in this IDD.

1. Consultative Committee for Space Data Systems (CCSDS),

Recommendation for Space Data System Standards; Advanced Orbiting Systems (AOS), Networks and Data Links: Architectural Specification, Blue Book, CCSDS 701.0-B-1, Issue 1, October 1992

2. NASA GSFC/MO&DSD, Landsat 7 Processing System (LPS) Functional and Performance Specification, Working Draft, 560-8FPS/0194, September 1994.

3. Martin Marietta Astro Space (MMAS), Landsat 7 System Data Format Control Book (DFCB), Revision A, Volume 4 - Wideband Data, 23007702, December 2, 1994.

4. NASA GSFC, Interface Control Document (ICD) between the Landsat 7 Ground Station (LGS) and the Landsat 7 Processing System (LPS), September 1994.

5. NASA GSFC, Interface Control Document between the EOSDIS Core System (ECS) and the Landsat 7 System, Working Draft, 194-219-SE1-003, August 1994. [Note: includes LPS-LP DAAC interface requirements].

6. NASA GSFC, Operational Agreement between the Landsat 7 Processing System and the Mission Operations Center (MOC), 1995 **(TBD)**.

7. NASA GSFC, Interface Control Document between the Landsat 7 Processing System and the Image Analysis System (IAS), 23007630, 1994.

8. Computer Sciences Corporation, Structured Systems Design Methodology, July 1989.

9. NASA GSFC/MO&DSD, Landsat 7 Processing System (LPS) Operations Concept, Draft, 560-3OCD/0194, September 1994.

10. NASA GSFC/MO&DSD, Landsat 7 Processing System (LPS) System Design Specification, Draft, 560-8SDS/0194, February 1995.

11. NASA GSFC/MO&DSD, Landsat 7 Processing System (LPS) Software Requirements Specifications, Draft, 560-8SWR/0195, April 1995.

12. NASA GSFC/MO&DSD, Landsat 7 Processing System (LPS). Preliminary Design Specification, Draft, June 30, 1995.

1.5 LPS Subsystem Interface Overview

Figures 1-1 and 1-2 show the high-level dataflows and the subsystems within LPS.

The Management and Control Subsystem (MACS) automatic data capture process invokes the Raw Data Capture Subsystem (RDCS) data capture program for each scheduled contact stored in the LPS database. The RDCS capture program captures raw wideband data and creates a raw capture file. The MACS invokes the RDCS data backup program to copy the raw capture file to removable media. The MACS invokes the RDCS data restage program to restage the raw capture file from removable media to disk.

The MACS top-level Level 0R program invokes the following Level 0R subsystem programs on operator command..

- Raw Data Processing Subsystem (RDPS)
- Major Frame Processing Subsystem (MFPS)
- Payload Correction Data Subsystem (PCDS)
- Image Data Processing Subsystem (IDPS)
- Landsat Data Transfer Subsystem (LDTS) Data Availability Notice (DAN) generation program.

The RDPS reads the raw capture file. It extracts and corrects Consultative Committee for Space Data Systems (CCSDS) Virtual Channel Data Units (VCDUs) from the raw capture file and appends quality annotations. It passes the annotated VCDUs to the MFPS. The RDPS stores quality and accounting information in the LPS database for retrieval by MACS report generation programs.

The MFPS extracts unpacked format Payload Correction Data (PCD) from each VCDU and passes these to the PCDS along with annotations indicating missing VCDUs and breaks in the data stream. The MFPS extracts ETM+ telemetry data from the annotated VCDUs and constructs ETM+ major frames. It passes the major frames to the IDPS. The MFPS identifies subinterval boundaries within the contact period and passes the subinterval boundary times to the IDPS, PCDS, and MACS. The MFPS stores quality and accounting information in the LPS database for retrieval by the MACS LPS Quality and Accounting Report generation program. The MFPS also stores metadata information in the LPS database for retrieval by MACS metadata generation functions. The MFPS stores output file names in the LPS database for retrieval by the LDTS DAN generation program.

The PCDS constructs PCD cycles (of four major frames) from the

unpacked format PCD received from the MFPS. The PCDS extracts Bands Present flags and passes these to the MACS for metadata generation. The PCDS extracts attitude and ephemeris points from the PCD and identifies the time at which the Landsat 7 spacecraft's view was closest to a Worldwide Reference System (WRS) scene center. The PCDS passes the scene center times to the IDPS. The PCDS stores metadata information in the LPS database for retrieval by the MACS metadata generation functions. The PCDS stores output file names in the LPS database for retrieval by the LDTS DAN generation program.

The IDPS constructs band files from the received ETM+ major frames. Using the ETM+ major frames and the scene center times received from the PCDS, the IDPS constructs a browse file for each full scene within a subinterval. For each full scene, the IDPS also performs automatic cloud cover assessment. The IDPS store metadata information in the database for retrieval by MACS metadata generation functions. The IDPS stores output file names in the LPS database for retrieval by the LDTS DAN generation program.

Upon successful completion of Level 0R processing by the RDPS, MFPS, PCDS, and IDPS, the MACS generates a metadata file for each subinterval within the contact. To do so, it retrieves the metadata information stored by each subsystem in the LPS database. The MACS then invokes the LDTS DAN generation program. The LDTS DAN generation program retrieves output file names from the LPS database to construct.

Before data capture, the RDCS data capture program will suspend all Level 0R processes. After data capture has complete, the RDCS data capture program causes the Level 0R processes to resume execution. The suspension and resumption is transparent to the Level 0R processes. However, the MACS top-level Level 0R program, which is the process group leader for the set of Level 0R processes processing a contact period, stores its process ID in the LPS database for retrieval by the RDCS.

The LPS operator can request three LPS reports via the MACS. The MACS invokes a RDCS report generation program to generate the Data Receive Summary. The MACS invokes a LDTS report generation program to generate the LPS Data Transfer Summary. In both cases, the MACS passes parameters specifying the report's scope. The MACS generates the LPS Quality and Accounting report by retrieving quality and accounting information collected by the RDPS and MFPS.

All subsystems report errors and status messages to the LPS

journal. On operator command, the MACS will display these messages either as they are written to the journal or by displaying the journal for operator browsing.

Figure 1-1 LPS Context Diagram

Figure 1-2 LPS Level-0 Diagram

2 Management And Control Subsystem

2.1 With Raw Data Capture Subsystem

2.1.1 Purpose of Interface

The purpose of the interface between MACS and RDCS is to provide MACS the capability of initiating RDCS, modifying the RDCS processing values stored in the database, and generating the Data Receive Summary report. Specifically the interface provides MACS the capability to control the following RDCS activities - raw data capture, raw data restage, raw data save, raw data and accounting file delete, and raw data report generation.

2.1.2 Description of Interaction

Database access routines, specifically SQL FORMS, provide the operator the capability of entering or modifying the LPS configuration parameter values (LPS_Configuration) stored in the database. The database validates these values using the domain integrity constraints in the database. MACS provides these parameters to RDCS via the database.

When MACS receives a directive from an operator to start RDCS, it directs RDCS to capture data from a specified contact period by passing to RDCS a contact schedule identifier, a data capture stop time, a string identifier, an isolate capture process flag, a suspend Level Zero processing flag, a suspend group Id and the capture mode. See section 2.1.3.

During a raw data capture session, the operator may wish to teminate an ongoing raw data capture. To stop an active raw data capture session, the operator selects the Stop Capture option from the Main Control Menu. MACS invokes rdc_StopCapture which teminates the active raw data capture session.

After the data from a specified contact period has been captured, the operator may invoke the raw data save process. To invoke the raw data save process, MACS invokes rdc_Save with the capture accounting file filename and the removable media device which stores the captured data onto a data storage media.

The operator may also request a raw data restage for a

specified contact period by selecting the appropriate option from the Main Control Menu. MACS, in turn invokes rdc_Restage with the removable media device to start the raw data restaging operation which copies the requested data from the data storage media back onto the disk for reprocessing.

After the data from the requested contact period has been captured, the operator may generate the Data Receive Summary report by selecting the Data Receive Summary report option from the Main Reports Menu screen. The ORACLE Report Writer is the interface between the operator and the database. The operator generates the Data Receive Summary report by invoking rdc_GenDataRcvSum-Report and inserting the contact sequence identifier into the Data Receive Summary option screen. MACS prints or displays the reports using system services.

2.1.3 Dataflow Structure

The following tables show the subsystem arguments and the data structures that provide information to the RDCS subsystem.

SRS Subsystem Arguments	PDS Subsystem Arguments	Type	Bytes
Contact_Sequence_Id	rdc_mainSchedIdArg	int	4
File_Version_No	rdc_mainSuspendGIDArg	int	4
Contact_Stop_Time	rdc_mainStopTimeArg	date	20
LPS_Hardware_String_Id	rdc_mainFileNameArg	int	4
Isolate_Flag	rdc_mainIsolateArg	int	4
Suspend_L0R_Process	rdc_mainSuspendArg	int	4
	rdc_mainCptrModeArg	char	1

2.1.4 IPC Mechanism(s)

The command line interface is the IPC mechanism used to pass appropriate parameters for raw data capture, raw data save, raw data restage, raw data and accounting file delete, and raw data receive summary reports. To initiate raw data capture,

MACS invokes rdc_Main with a contact schedule identifier, the capture mode, a stop time, a string identifier, a suspend process flag and group Id, and an isolate capture flag. To initiate raw data save, MACS invokes rdc_Save with the accounting filename and the device name. To initiate raw data restage, MAC invokes rdc_Restage with a device name. To stop an active raw data capture, MACS invokes rdc_StopCapture to terminate the active capture. To stop an active raw data save and an active raw data restage, MACS invokes rdc_StopSaveRestage with an identifier whether to stop save or restage. To initiate the raw data summary report, MACS invokes rdc_GenDataRcvSumReport with a contact sequence identifier.

2.2 With Raw Data Processing Subsystem

2.2.1 Purpose of Interface

The purpose of the interface between the Management and Control Subsystem (MACS) and the Raw Data Processing Subsystem (RDPS) is to provide MACS the capability of invoking RDPS, modifying the RDPS processing values stored in the database, and generating the Return Link QA report.

2.2.2 Description of Interaction

Database access routines provide the operator the capability of entering or modifying the CCSDS parameters (RDP_CCSDS_Parms) and the data threshold values (RDP_Thresholds) in the database via ORACLE forms. The database validates these values using the the domain integrity constraints in the database. MACS provides these parameters and thresholds to RDPS via the database.

After RDCS captures a contact period or the MACS completes a restaging operation, MACS directs RDPS to process the data from a specified contact period by passing to RDPS a contact sequence identifier, the raw wideband data file name, a file version number, and a flag indicating whether or not BCH correction must be performed. See section 2.2.3.

After the data from the requested contact period has been processed, the operator may generate the Return Link QA report by selecting the Return Link QA report option from the Main Reports Menu screen. The ORACLE Report Writer is the interface between the operator and the database. The operator generates the Return Link QA report by providing the contact sequence identifier and the file version number in the appropriate fields on the Return Link QA option screen. MACS prints or displays the reports using system services.

2.2.3 Dataflow Structures

The following tables show the subsystem arguments and the data structures that provide information to the RDPS subsystem.

SRS Subsystem Arguments	DDS Subsystem Arguments	Type	Bytes
Contact_Sequence_Id	mac_ContSeqId	int	4
N/A	mac_Filename	char	<=256
File_Version_No	mac_FileVerNo	int	4
PCD_Processing_Flag	mac_BCHFlag	char	1

SRS Data Structure	DDS Data Structure	Type	Bytes
Ann_CADU	lpsAnnotatedCADUStruct	struct	1068
Sync	lpsAnnotatedCADUStruct->Sync	char	4
VCDU_Hdr_Bytes	lpsAnnotatedCADUStruct->VCDU_Hdr_Bytes	char	8
[VCDU_Data	lpsAnnotatedCADUStruct->VCDU_Data	char	992
BCH_Corrected_Data]	lpsAnnotatedCADUStruct-> BCH_Corrected_Data	char	992
VCDU_Trailer	lpsAnnotatedCADUStruct->VCDU_Trailer	char	2
Sync_Annotation	lpsAnnotatedCADU Struct-> Sync_Annotation	struct	5
CADU_File_Offset_In_ Bytes	lpsAnnotatedCADU Struct-> CADU_File_Offset_ In_Bytea	lpsInt64 Type	8
CADU_Bit_Slip_Flag	lpsAnnotatedCADU Struct-> Sync_Annotation. CADU_Bit_Slip	lpsInt8 Type	1
CADU_Polarity_Flag	lpsAnnotatedCADU Struct-> Sync_Annotation. CADU_Polarity_Flag	lpsInt8 Type	1

CADU_Sync_Error_Flag	lpsAnnotatedCADU Struct-> Sync_Annotation. CADU_Sync_Error_ Flag	lpsInt8 Type	1
CADU_Flywheel_Flag	lpsAnnotatedCADU Struct-> Sync_Annotation. CADU_Flywheel_ Flag	lpsInt8 Type	1
End_Of_Contact_Flag	lpsAnnotatedCADU Struct-> Sync_Annotation. End_Of_Contact_ Flag	lpsInt8 Type	1
CRC_Annotation	lpsAnnotatedCADUStruct->CRC_Annotation	lpsInt8 Type	1
RS_Annotation	lpsAnnotatedCADUStruct->RS_Annotation	enum	1
BCH_Annotation	lpsAnnotatedCADUStruct->BCH_Annotation	struct	4
BCH_Mission_Quality_ Indicator	lpsAnnotatedCADUStruct-> BCH_Annotation. BCH_Mission_Quality_Indicator	lpsInt8 Type	1
BCH_Mission_Bits_ Corrected_Count	lpsAnnotatedCADUStruct-> BCH_Annotation. BCH_Misson_Bits_ Corrected_Count	lpsInt8 Type	1
BCH_Pointer_Quality_ Indicator	lpsAnnotatedCADUStruct-> BCH_Annotation.BCH_Pointer_Quality_ Indicator	lpsInt8 Type	1

BCH_Pointer_Bits_ Corrected_Count	lpsAnnotatedCADUStruct-> BCH_Annotation. BCH_Pointer_Bits_ Corrected_Count	lpsInt8 Type	1
VCID_Change_Flag	lpsAnnotatedCADUStruct-> VCID_Change_Flag	Boolean	1
Fill_CADU_Flag	lpsAnnotatedCADUStruct->Fill_CADU_Flag	Boolean	1

2.2.4 IPC Mechanism(s)

The command line interface is the IPC mechanism used to pass appropriate parameters to the raw data processing subsystem. To initiate RDPS, MACS invokes rdp_Main with contact sequence identifier, the raw wideband data file name, a file version number, and a BCH correction flag. To initiate the database entry or modification process, MACS invokes mac_db_LPSThres and/or mac_db_LPSParms. The operator inserts the information directly into the fields on these screens. When the operator completes the insertions, MACS stores the information in the appropriate database tables. To invoke the Return Link QA report processor, MACS invokes mac_db_ReturnLinkQA with a contact sequence identifier and a file version number.

2.3 With Major Frame Processing Subsystem

2.3.1 Purpose of Interface

The purpose of the interface between the Management and Control Subsystem (MACS) and the Major Frame Processing Subsystem (MFPS) is to provide MACS the capability of invoking MFPS, modifying the MFPS processing values stored in the database, and generating the Level 0R QA report.

2.3.2 Description of Interaction

Database access routines provide the operator the capability of entering or modifying the MFPS parameters (MFP_Parms) and the data threshold values (MFP_Thresholds) in the database via ORACLE forms. See section 2.3.3 for details. The database validates these values using the the domain integrity constraints in the database. MACS provides these parameters and thresholds to MFPS via the database.

After RDCS captures the data associated with the requested contact period, MACS directs MFPS to begin major frame processing by invoking mfp_Main with a contact sequence identifier and a file version number. See section 2.3.3.

After the telemetry data associated with the requested contact period has been processed, the operator may generate the Level 0R QA report by selecting the Level 0R QA report option from the Main Reports Menu screen. The ORACLE Report Writer is the interface between the operator and the database. The operator requests the Level 0R QA report by invoking mac_db_Level0RQA with a contact sequence identifier and a file version number. MACS prints or displays the reports using system services.

2.3.3 Dataflow Structure

The following tables show the subsystem arguments and the data structures that provide information to the MFPS subsystem.

SRS Subsystem Arguments	PDS Subsystem Arguments	Type	Bytes
Contact_Sequence_Id	lps_mainContactIdArg	int	4
File_Version_Number	lps_mainFileVerNumArg	int	4

SRS Data Structures	PDS Data Structures	Type	Bytes
MFP_Directive	MFP_Directive		
MFP_Parms	mfp_parms		
Sensor_Alignment_Info	mfp_sensorAlignInfo	struct	78
Fill_Value	mfp_fillValue	byte	1
Sub_Intv_Delta	mfp_subIntvDelta	lpsTime Struct	96
Mjf_Data_Rate	mfp_dataPeriod	lpsTime Struct	96
Max_Alignment_Value	mfp_maxAlignValue	short	2
Time_Range_Tol	mfp_timeRangeTol	lpsTime Struct	96
Part_Mnf_Tol	mfp_partMnfTol	short	2
Maj_Vote_Tol	mfp_majVoteTol	short	2
	mfp_EolTol	int	4
	mfp_MajSyncTol	short	2
(from MFP_Thresholds - Scan_Dir_Thr)	mfp_maxCaduCnt	int	4
MFP_Thresholds	mfp_thres		
Mjf_CADU_Seq_Err_Thr	mfp_caduSeqErrThr	short	2
Scan_Dir_Thr	(moved to MFP_Parameters)	short	2
Sync_Thr	mfp_caduSyncThr	short	2
Mnf_Ctr_Thr	mfp_mnfCtrThr	short	2
Eol_Thr	mfp_eolErrThr	short	2
Tc_Thr	mfp_tcErrThr	short	2
Full_Mjf_Thr	mfp_fullFillThr	short	2

Part_Mjf_Thr	mfp_partFillThr	short	2
MFP_Rpt_L0R_QA_Drct			
Contact_Sequence_Id	lps_mainContactIdArg	int	4
LDP_Hardware_String Id			
LGS_Channel_Id			
Contact_Start_Time			
Contact_Stop_Time			

2.3.4 IPC Mechanism(s)

The command line interface is the IPC mechanism used to pass appropriate parameters to the major frame processing subsystem. To initiate MFPS, MACS invokes mfp_Main with a contact sequence identifier and a file version number. To initiate the MFPS database entry/update operation, MACS invokes mac_MainLPSParms and/or mac_MainLPSThres. The operator inserts the information into fields within the screen. When the operator completes the insertion process, MACS updates the appropriate database tables. To initiate the Level 0R QA report process, MACS invokes mac_db_Level0RQA with a contact sequence identifier and a file version number.

2.4 With Image Data Processing Subsystem

2.4.1 Purpose of Interface

The purpose of the interface between the Management and Control Subsystem (MACS) and the Image Data Processing Subsystem (IDPS) is to provide MACS the capability of invoking IDPS and modifying the IDPS processing values stored in the database.

The MACS invokes the IDPS and provides the contact sequence identifier (mac_ContSeqId) and the file version number (mac_FileVerNo) as arguments.

The MACS also provides the IDP band parameters (IDP_Band_Parms) that are needed for image data processing. The parameters are stored in the LPS database and can be modified by the operator.

2.4.2 Description of Interaction

Database access routines provide the operator the capability of entering or modifying the IDPS bandwidth parameters (IDP_Band_Parms) into the database via ORACLE forms. See section 2.4.3 for details. The database validates these values using the domain integrity constraints in the LPS database. The MACS provides these parameters to IDPS via the LPS database.

Upon receipt of a start processing directive, the MACS invokes the IDPS, with the appropriate subsystem arguments, using Unix system call(s). The IDPS uses the argument in the processing of image data associated with the specified contact sequence identification.

2.4.3 Dataflow Structure

The following tables show the subsystem arguments and the data structures that provide information to the IDPS subsystem.

SRS Subsystem Arguments	PDS Subsystem Arguments	Type	Bytes
Contact_Sequence_Id	mac_ContSeqId	int	4
File_Version_Number	mac_FileVerNo	int	4

SRS Data Structures	PDS Data Structures	Type	Bytes
IDP_Directive	IDP_Directive		
IDP_Band_Parms	IDP_Band_Parms		
	Mono	short	2
(Multi1)	Multi1	short	2
(Multi2)	Multi2	short	2
(Multi3)	Multi3	short	2
(Fmt)			
(Subs)	Subs	short	2
(Wave)	Wave	short	2
(CCA_Method)	CCA_Method	string	TBD
(CCA_Ratio)	CCA_Ratio	short	2

2.4.4 IPC Mechanism(s)

The LPS database is the IPC mechanism used to provide the IDP_Band_Parms to the IDPS.

The UNIX operating system is the IPC mechanism used to invoke the IDPS. The Subsystem arguments are passed to the IDPS as command line arguments at the time of invocation.

2.5 With Payload Correction Data Processing Subsystem

2.5.1 Purpose of Interface

The purpose of the interface between the Management and Control Subsystem (MACS) and the Payload Correction Data Processing Subsystem (PCDS) is to provide MACS the capability of invoking PCDS and modifying the PCDS processing values stored in the database.

The MACS invokes the PCDS and provides the contact sequence identifier (mac_ContSeqId), file version number (mac_FileVerNo), and the processing status flag (mac_PCDProcFlag) as arguments.

The MACS also provides the PCD parameters (PCD_Parms) and thresholds (PCD_Thresholds) that are needed for payload correction data processing. The parameters and thresholds are stored in the LPS database and can be modified by the operator.

2.5.2 Description of Interaction

Database access routines provide the operator the capability of entering or modifying the PCDS scene identification setup parameters (PCD_Parms) and the data threshold values (PCD_Thresholds) in the database via ORACLE forms. See section 2.5.3 for details. The LPS database validates these values using the the domain integrity constraints in the database. MACS provides these parameters and thresholds to PCDS via the LPS database.

Upon receipt of a start processing directive, the MACS invokes the PCDS, with the appropriate subsystem arguments, using Unix system call(s). The PCDS uses the arguments in the processing of payload correction data associated with the specified contact sequence identification.

2.5.3 Dataflow Structure

The following tables show the subsystem arguments and the data structures that provide information to the PCDS subsystem.

SRS Subsystem Arguments	DDS Subsystem Arguments	Type	Bytes
Contact_Sequence_Id	mac_ContSeqId	int	4
File_Version_No	mac_FileVerNo	int	4

SRS Data Structure	DDS Data Structure	Type	Bytes
Scene_Parms	Scene_Parms		
(ETM_Plus_To_Body_Trans)	ETM_Body_Trans_Matrix_11	double	8
	ETM_Body_Trans_Matrix_12	double	8
	ETM_Body_Trans_Matrix_13	double	8
	ETM_Body_Trans_Matrix_21	double	8
	ETM_Body_Trans_Matrix_22	double	8
	ETM_Body_Trans_Matrix_23	double	8
	ETM_Body_Trans_Matrix_31	double	8
	ETM_Body_Trans_Matrix_32	double	8
	ETM_Body_Trans_Matrix_33	double	8
(Mission_Start_Time)	Mission_Start_TIme	double	8
(Time_Per_Orbit)	Time_Per_Orbit	double	8
(ETM_Plus_LOS_x)	ETM_Plus_LOS_X	float	4
(ETM_Plus_LOS_y)	ETM_Plus_LOS_Y	float	4
(ETM_Plus_LOS_z)	ETM_Plus_LOS_Z	float	4
(Semi_Major_Axis)	Semi_Major_Axis	float	4
(Semi_Minor_Axis)	Semi_Minor_Axis	float	4
PCD_Parms	PCD_Parms		
(PCD_Frame_Fill)	PCD_Frame_Fill_Value	int	4

WRS_Parms	WRS_Parms		
({WRS_Row_Nominal	WRS_Row_Nominal	short	2
Latitude}	Scene_Center_Latitude	float	4
{WRS_Path_Nominal	WRS_Path_Nominal	short	2
Longitude})	Scene_Center_Longitude	float	4
PCD_Thres	PCD_Thres		
(Ephem_Position_Upper)	Ephem_Position_Upper	float	4
(Ephem_Position_Lower)	Ephem_Position_Lower	float	4
(Ephem_Velocity_Upper)	Ephem_Velocity_Upper	float	4
(Ephem_Velocity_Lower)	Ephen_Velocity_Lower	float	4
(Att_Lower_Bounds)	Attitude_Quaternion_Tol	float	4
(Att_Upper_Bounds)			
(Num_Missing_Data_Words)	Num_Mission_Data_Words	int	4
(Num_Failed_Votes)	Num_Failes_Votes	int	4

2.5.4 IPC Mechanism(s)

The LPS database is the IPC mechanism used to provide the PCD_Parms, PCD_Thres, Scene_Parms, and WRS_Parms to the PCDS.

The UNIX operating system is the IPC mechanism used to invoke the PCDS. The Subsystem arguments are passed to the PCDS as command line arguments at the time of invocation.

2.6 With LPS Data Transfer Subsystem

2.6.1 Purpose of Interface

The purpose of the interface between the Management and Control Subsystem (MACS) and the LPS Data Transfer Subsystem (LDTS) is to provide MACS the capability of controlling the following activities - to send or resend data availability notifications (DAN's) to the LP DAAC, to delete or retain LPS output file sets, to enable or disable the transfer of DAN's to the LP DAAC, to check for DAN's which have timed out, to invoke the LDTS DDN server, and to generate file transfer summary reports.

2.6.2 Description of Interaction

Once the RDPS, MFPS, IDPS, and PCDS inform MACS that the data associated with a contact period has been processed, MACS invokes the LDTS process ldt_SendDAN with a contact sequence identifier, a file version number, and a request type ("SEND" or "RESEND"). This process generates a DAN and sends it to the LP DAAC if DAN transfer is enabled. If DAN transfer is disabled, the operator is informed that the DAN has been suspended because DAN transfer is not currently enabled.

An operator may retain or delete files in a file set. To retain or delete files, MACS invokes ldt_RetainFiles or ldt_DeleteFiles with a contact sequence identifier and a file version number. Additionally, ldt_DeleteFiles takes an override retention argument and an "only delete ingested files" argument. Argument values are specified by the operator inserting information into fields on the screen.

The operator can enable or disable DAN transfer to the LP DAAC by invoking mac_db_EnableDisableTrans with a transfer state argument of "Enable" or Disable".

On each LPS string, the LDTS process ldt_CheckTimeouts is executed several times daily as a unix cron job. As a result, a list of any DAN's which have timed out appears in the LPS journal as a list of contact sequence identifier and file version number pairs.

When LPS processing begins, MACS must activate the LDTS DDN server process ldt_RcvDDN. ldt_RcvDDN watches a well-known port for incoming socket connections from the LP DAAC. Once

established, each non-concurrent socket is used to accept Data Delivery Notices (DDN's) from the LP DAAC. ldt_RcvDDN runs "forever" unless told terminate by MACS.

At any time the operator may generate a File Transfer Summary report by selecting the File Transfer Summary report option from the Main Reports Menu screen. When the operator requests a File Transfer Summary report, MACS invokes ldt_GenFTS with a report start time, a report stop time. The start and stop times define the interval of time for which LDTS will generate report information. MACS prints or displays the report using system services.

2.6.3 Dataflow Structure

The following tables show the subsystem arguments and the data structures that provide information to the LDTS subsystem.

SRS Subsystem Arguments	DDS Subsystem Arguments	Type	Bytes
Contact_Sequence_Id	mac_ContSeqId	int	4
File_Version_No	mac_FileVerNo	int	4
Request_Type	mac_ReqType	char	6
Transfer_State	mac_XferState	char	7
Report_Start_Time	mac_RptStartTime	struct	20
Report_Stop_Time	mac_RptStopTime	struct	20
Transfer_File_Name	mac_FileName	char	50

SRS Data Structure	DDS Data Structure	Type	Bytes
LDT Output Files Table	LPS File Info Table		
Contact_Sequence_Id	Contact_Sequence_Id	int	4
File_Version_No	File_Version_No	int	4
File_Name	File_Name	char	50
File_Type	File_Type	char	10

SRS Data Structure	DDS Data Structure	Type	Bytes

LDT Output Files	LDT File Set Info		
n/a	File_Set_Status	int	2

2.6.4 IPC Mechanism(s)

The command line interface is the IPC mechanism used to pass appropriate parameters to the LPS data transfer subsystem. To initiate the send or resend DAN process, MACS invokes ldt_SendDAN with a list of contact sequence identifier / file version number pairs, and a request type. To initiate the retain or delete files process, MACS invokes ldt_RetainFiles or ldt_DeleteFiles with a contact sequence identifierand file version number, and, additionally, for ldt_DeleteFiles, the override and ingested boolean arguments. To initiate the enable or disable process, MACS invokes mac_db_EnableDisableTrans with the requested transfer state. To access the output of the ldt_CheckTimeouts process, the LPS Journal is used. To initiate the LDTS DDN server process, MACS invokes ldt_RcvDDN when LPS processing begins. To generate File Transfer Summary reports, MACS invokes ldt_GenFTS with a report start time, a report stop time, and a file name for the report.

3 Raw Data Capture Subsystem

3.1 With Raw Data Processing Subsystem

3.1.1 Purpose of Interface

This interface provides the RDPS with the captured raw wideband data (Raw_WB_Data) associated with a contact period for processing.

3.1.2 Description of Interaction

RDCS stores received raw wideband data (Raw_Data_Byte_Stream) in files (Raw_WB_Sets). The file names are produced from the Contact_Start_Time, Contact_Stop_Time and the LPS_Hardware_String_Id with the format: YY_DDD_HH:MM:SS_XX..X.RDC_CPTR.EXT where XX..X is the capture string name. If the raw data is test data, the dot extension is RDC_TEST_EXT. These file names are stored in the database (RDC_Acct) and retrieved by RDPS.

3.1.3 Dataflow Structure

The following shows the format of the raw data files (Raw_WB_Sets).

Data Structure (SRS)	Data Structure (PDS)	Type	Bytes
Raw_WB_Sets		char	TBD
Contact_Id		struct	TBD
LPS_Hardware_String_Id	rdc_mainStringIdArg	int	4
LGS_Channel_Id		char	TBD
Contact_Start_Time		time	TBD
Contact_Stop_Time	rdc_mainStopTimeArg	time	TBD
Raw_Data_Byte_Stream		byte	TBD

3.1.4 IPC Mechanism(s)

Unix files on a shared system disk will be used as the IPC mechanism between the RDCS and the RDPS for the Raw_WB_Sets. The database will be used as the IPC mechanism between the RDCS and the RDPS for the file names of the Raw_WB_Sets.

3.2 With Management And Control Subsystem

3.2.1 Purpose of Interface

This interface allows the RDCS to report status and accounting information to the MACS.

3.2.2 Description of Interaction

Raw wideband data capture begins with the invocation of RDCS by the MACS. RDCS retrieves configuration information from the database (LPS_Configuration) which has been generated by the MACS.

RDCS generates the capture accounting table (RDC_Acct) in the database which is accessed by both MACS and RDPS.

Status messages for printing or display are sent to the MACS by the global routine lps_LogMessage.

3.2.3 Dataflow Structure

The following shows the dataflow structure used to pass the accounting information to the MACS.

Data Structure (SRS)	Data Structure (PDS)	Type	Bytes
RDC_Acct	RDC_Acct	struct	TBD
	rdc_mainCptrModeArg	char	
	rdc_mainFileNameArg	char string	
	rdc_mainIsolateArg	boolean	

	rdc_mainSchedIdArg	int	
	rdc_mainStopTimeArg	int	TBD
	rdc_mainSuspendArg	boolean	
	rdc_mainSuspendGIDArg	int	TBD

3.2.4 IPC Mechanism(s)

The global routine lps_LogMessage will be the IPC mechanism between RDCS and MACS for status messages. The database will be the IPC mechanism between RDCS and MACS for the accounting information (RDC_Acct).

4 Raw Data Processing Subsystem

4.1 With Major Frame Processing Subsystem

4.1.1 Purpose of Interface

The interface between the RDPS and the MFPS provides the MFPS with the annotated VCDUs (Ann_VCDU), which have been identified, verified and, in some cases, corrected by RDPS.

4.1.2 Description of Interaction

As the processing of each CADU has been completed by RDPS, the annotated VCDU (Ann_VCDU) is made available to the MFPS by placing it into the shared memory. RDPS processes one block at a time. The annotated VCDU is defined in section 4.1.3 below.

4.1.3 Data flow Structure

The following shows the dataflow structures used to pass the information to the MFPS subsystem.

Data Entity	DDS Data Structure	Type	Bytes
Ann_CADU	lpsAnnotatedCADUStruct	struct	1068
Sync	lpsAnnotatedCADUStruct -> Sync	char	4
VCDU_Hdr_Bytes	lpsAnnotatedCADUStruct -> VCDU_Hdr_Bytes	char	8
[VCDU_Data |	lpsAnnotatedCADUStruct -> VCDU_Data	char	992
BCH_Corrected_Data]	lpsAnnotatedCADUStruct -> BCH_Corrected_Data	char	992
VCDU_Trailer	lpsAnnotatedCADUStruct -> VCDU_Trailer	char	2

Sync_Annotation	lpsAnnotatedCADUStruct -> Sync_Annotation	struct	5
CADU_File_Offset_In_ Bytes	lpsAnnotatedCADUStruct -> Sync_Annotation.CADU_ File_Offset_In_Bytes	lpsInt64 Type	8
CADU_Bit_Slip_Flag	lpsAnnotatedCADUStruct -> Sync_Annotation.CADU_ Bit_Slip	lpsUInt 8Type	1
CADU_Polarity_Flag	lpsAnnotatedCADUStruct -> Sync_Annotation.CADU_ Polarity_Flag	lpsUint8 Type	1
CADU_Sync_Error_Flag	lpsAnnotatedCADUStruct -> Sync_Annotation.CADU_ Sync_Error_Flag	lpsUint8 Type	1
CADU_Flywheel_Flag	lpsAnnotatedCADUStruct -> Sync_Annotation.CADU_ Flywheel_Flag	lpsUint8 Type	1
End_Of_Contact_Flag	lpsAnnotatedCADUStruct -> Sync_Annotation.End_Of _ Contact_Flag	lpsUint8 Type	1
CRC_Annotation	lpsAnnotatedCADUStruct -> CRC_Annotation	lpsUint8 Type	1
RS_Annotation	lpsAnnotatedCADUStruct -> RS_Annotation	enum	1
BCH_Annotation	lpsAnnotatedCADUStruct -> BCH_Annotation	struct	4

BCH_Mission_Quality_ Indicator	lpsAnnotatedCADUStruct -> BCH_Annotation.BCH_Mission_Quality_ Indicator	lpsUInt8Type	1
BCH_Mission_Bits_ Corrected_Count	lpsAnnotatedCADUStruct -> BCH_Annotation.BCH_Mission_Bits_ Corrected_Count	lpsUInt8Type	1
BCH_Pointer_Quality_ Indicator	lpsAnnotatedCADUStruct -> BCH_Annotation.BCH_Pointer_Quality_ Indicator	lpsUInt8Type	1
BCH_Pointer_Bits_ Corrected_Count	lpsAnnotatedCADUStruct -> BCH_Annotation.BCH_Pointer_Bits_ Corrected_Count	lpsUInt8Type	1
VCID_Change_Flag	lpsAnnotatedCADUStruct -> VCID_Change_Flag	Boolean	1
Fill_CADU_Flag	lpsAnnotatedCADUStruct -> Fill_CADU_Flag	Boolean	1

4.1.4 IPC Mechanism(s)

Shared memory is used to pass the annotated CADUs from RDPS to MFPS to reduce data copying. The shared memory will be arranged so that each block of memory contains TBD VCDUs. Shared memory is accessed using global shared memory access routines.

4.2 With Management And Control Subsystem

4.2.1 Purpose of Interface

RDPS interfaces with the MACS for status and error reporting and return link report generation.

4.2.2 Description of Interaction

As RDPS is processing the raw wideband data, the status and error messages will be handled by calling the lps_LogMessage global routine (see section XXX for details). Quality and accounting information is stored in the database by contact id to be used to generate the return link quality and accounting report. The return link statistics generated by RDPS will be sent to the MACS using UNIX system services.

4.2.3 Dataflow Structure

SRS Data Structure	DDS Data Structure	Type	Bytes
Contact_Id	rdp_mainContactIdArg	int	2

SRS Data Structure	DDS Data Structure	Type	Bytes
N/A	rdp_mainVersionIdArg	int	2

SRS Data Structure	DDS Data Structure	Type	Bytes
N/A	rdp_mainMustBCHArg	char	1

SRS Data Structure	DDS Data Structure	Type	Bytes
N/A	rdp_mainDataFilenameArg	char [256]	<= 255

The data quality and accounting information (RDP_Acct) is made available to the MACS as a database file and is defined as follows:

SRS Data Structure	DDS Data Structure	Type	Bytes
RDP_Acct	RDP_Acct	struct	80
rdp_mainContactIDArg	Contact_Sequence_Id	lpsUInt32 Type	4
rdp_mainVersionArg.	File_Version_Number	lpsUInt32 Type	4
rdp_mainCCSDSIndexArg	CCSDS_Parms_Id	lpsUInt32 Type	4
rdp_mainAcctInfoArg.CADU_ Polarity	Inverted_CADU_Cnt	lpsUInt32 Type	4
	Polarity_Change_Cnt	lpsUInt32 Type	4
rdp_mainAcctInfoArg.CADU_Bit_ Slip	CADU_Bit_Slip_Cnt	lpsUInt32 Type	4
rdp_mainAcctInfoArg.CADU_Sync_Error_Count	CADU_Sync_Err_Cnt	lpsUInt32Type	4
rdp_mainAcctInfoArg.CADU_Rcv_ Count	CADU_Rcv_Cnt	lpsUInt32 Type	4
rdp_mainAcctInfoArg.CADU_ Flywheel_Count	CADU_Flywheel_Cnt	lpsUInt32 Type	4
	Fill_CADU_Cnt	lpsUInt32 Type	4
rdp_mainAcctInfoArg.CRC_Error_Count	CADU_CRC_Err_Cnt	lpsUInt32 Type	4
	VCDU_Header1_Correctable_ Err_Cnt	lpsUInt32 Type	4

	VCDU_Header2_Correctable_ Err_Cnt	lpsUInt32 Type	4
rdp_mainAcctInfoArg.VCDU_ Header_Uncorrectable_Error_ Count	VCDU_Header_Uncorrectabl e_ Err_Cnt	lpsUInt32 Type	4
rdp_mainAcctInfoArg.BCH_Dat a_ Corrected_Error_Count	BCH_Data_Corrected_Err_Cn t	lpsUInt32 Type	4
rdp_mainAcctInfoArg.BCH_Dat aUncorrected_Error_Count	BCH_Data_Uncorrected_Err Cnt	lpsUInt32 Type	4
	BCH_Data_Corrected_ Bits_Cnt	lpsUInt32 Type	4
rdp_mainAcctInfoArg.BCH_Poi nter_Corrected_Error_Count	BCH_Pointer_Corrected_Err _ Cnt	lpsUInt32 Type	4
rdp_mainAcctInfoArg.BCH_Poi nter_Uncorrected_Error_Count	BCH_Pointer_Uncorrected_ Err_Cnt	lpsUInt32 Type	4
	BCH_Pointer_Corrected_ Bits_Cnt	lpsUInt32 Type	4

4.2.4 IPC Mechanism(s)

Status and error messages are handled by the Unix system facilities (see section XXX for detail). The database will be used to generate the return link quality and accounting report (Report_RDP_Return_Link_QA) through Oracle Report Writer.

5 Major Frame Processing Subsystem

5.1 With Image Data Processing Subsystem

5.1.1 Purpose of Interface

This interface allows the IDPS to receive the aligned bands from the MFPS on a major frame basis.

5.1.2 Description of Interaction

MFPS writes to the shared memory buffer the following information: the subinterval id, the status information, the major frame time, band 6 aligned data and the aligned band data for either format 1 or format 2. MFPS writes to a block every major frame with the exception of the subinterval id. Since this changes only once per subinterval, it is written to the block only at the beginning of a subinterval. Status information (Status_Info) consists of the system status and band gains from the PCD/Status data of the VCDU data zone. The major frame time will be the actual time extracted from the ETM+ major frame, or if that is unreliable, then an estimated time will be substituted. The sensor alignment value for the aligned band data differs for each band and each odd/even detector. For sake of clarification, sample values from one of the band/detector combinations are listed for sensor alignment in the table.

Enough blocks will exists to allow MFPS to write to a block while IDPS reads from another.

The subinterval information is placed into the database. MFPS enters the Sub_Intv information twice on a subinterval basis - once at the beginning of a subinterval and again when the subinterval stop time is identified. The Sub_Intv includes the contact id (Contact_Sequence_Id), the file version number (File_Version_Num), the subinterval id (Sub_Intv_Sequence_Id), the subinterval start time (MF_Start_Time), the subinterval stop time (MF_Stop_Time), and the virtual channel id (VCID). IDPS retrieves the Sub_Intv by querying the LPS database and keying on the Sub_Intv_Sequence_Id it receives from the shared memory block.

5.1.3 Dataflow Structure

The following shows the dataflow structures used to pass the information to the IDPS subsystem.

Data Structure (SRS)	Data Structure (DDS)	Type	Bytes
Aligned_Bands	lps_AlignedBands	byte	~552K
Sub_Intv_Id	lps_subIntvId	int	4
Status_Info	mfp_statusInfo	byte	2
System_Status		byte	1
Band_Gains		byte	1
Major_Frame_Time	mfp_mainMjfTimeArg	Time	25
8 {	8{	byte	~31K
0{Fill_Value} Max_Alignment_Value	0{mfp_fillValue} mfp_sensorAlignInfo	byte	15
Band6_Det_Data	mfp_band6Data	byte	~3.7K
0{Fill_Value} Max_Alignment_Value	0{mfp_fillValue} [mfp_maxAlignValue - mfp_sensorAlignInfo]	byte	179
} 8	} 8		
[Fmt1_Align_Data |	[mfp_fmt1AlignData|	struct	~521K
5 { 16 {	5 { 16 {		
0{Fill_Value} Max_Alignment_Value	0{mfp_fillValue} mfp_sensorAlignInfo	char	188
Band_Det_Data	mfp_bandDetData	byte	~6.3K
0{Fill_Value} Max_Alignment_Value	0{mfp_fillValue} [mfp_maxAlignValue - mfp_sensorAlignInfo]	char	6
} 16 } 5	} 16 } 5		
Fmt2_Align_Data]	mfp_fmt2AlignData]	struct	~521K
16 {	16 {	byte	~104
0{Fill_Value} Max_Alignment_Value	0{mfp_fillValue} mfp_sensorAlignInfo	byte	42

Band_Det_Data	mfp_bandDetData	byte	~6.3K
0{Fill_Value} Max_Alignment_Value	0{mfp_fillValue} [mfp_maxAlignValue - mfp_sensorAlignInfo]	byte	152
} 16	} 16		
64 {	64 {		~416K
0{Fill_Value} Max_Alignment_Value	0{mfp_fillValue} mfp_sensorAlignInfo	byte	194*
Pan_Det_Data	mfp_panDetData	byte	~6.3K
0{Fill_Value} Max_Alignment_Value	0{mfp_fillValue} [mfp_maxAlignValue - mfp_sensorAlignInfo]	byte	0*
} 64	} 64		

* The maximum alignment for one is not known, but this is an estimated value based on the total alignment value used by the other bands.

The following shows the contents of the subinterval information.

Data Structure	**Type**	**Bytes**
Sub_Intv	struct	63
Contact_Sequence_Id	int	4
File_Version_Num	int	4
Sub_Intv_Sequence_Id	int	4
MF_Start_Time	char	25
MF_Stop_Time	char	25
VCID	byte	1

5.1.4 IPC Mechanism(s)

Due to the large size of the Aligned_Bands, the IPC mechanism will be shared memory. Shared memory will be arranged so that each block of memory contains one Aligned_Bands. Access to the shared memory is done using global shared memory access routines. The database is used for the storage of information about each subinterval. This was chosen in order

to allow convenient access by all subsystems to the subinterval information.

5.2 With Payload Correction Data Processing Subsystem

5.2.1 Purpose of Interface

This interface allows PCDS to receive, from MFPS, the PCD bytes and information about the current contact period and subinterval on the basis of a MFPS VCDU set.

5.2.2 Description of Interaction

MFPS sends the PCD data to the PCDS subsystem on the basis of a MFPS VCDU set. MFPS VCDU sets are of two types: one type contains the VCDUs belonging to an identifiable major frame and the other type does not belong to a major frame and is written to a trouble file. PCD data is extracted from both types of sets. PCD data from each set is extracted and written to a block of shared memory. The PCDS reads the same data from the shared memory block. The MFPS writes the PCD data a block at a time to shared memory for PCDS. The shared memory block size is the size of the lps_mainPcdInfoStruct structure, described in section 5.2.3 below. The shared memory block is capable of storing PCD data from the largest MFPS VCDU set. When the MFPS is ready to write the PCD data to the shared memory, it locks an available shared memory block for writing the PCD data, writes the data and then unlocks the block so the PCDS can read it. After the PCDS has read a shared memory block, the block again becomes available to MFPS for writing more PCD data.

5.2.3 Dataflow Structure

The following shows the dataflow structure used to pass the information to PCDS. As is obvious, the data structure used to send data to the PCDS has been modified during the design phases, since the SRS phase.

Data Structure (SRS)	Data Structure (DDS)	Type	Bytes
PCD_Info	lps_mainPcdInfo	struct	5860
	lps_pcdCnt	int	4

Sub_Intv_Id	lps_subIntvId	int	4
End_Of_Contact_Flag	lps_eocFlag	short	2
	650{lps_vcduInfo	struct	9
PCD_Bytes	lps_pcdBytes	Byte	4
Num_Missing_VCDUs	lps_missingVcduCnt	Int	4
	lps_vcid	Bit	6 Bits
	lps_vcidChgFlag	Bit	1 Bit
	lps_dataBrk	Bit	1 Bit
	}650		

The structure lps_vcduInfo contains the data sent to PCDS from each VCDU. The field lps _pcdBytes contains the four PCD bytes extracted from the VCDU. Field lps_vcid contains the VCID of the VCDU, when lps_vcidChgFlag is TRUE; otherwise this field is undefined. Field lps_missingVcduCnt contains the number of VCDUs missing before the VCDU, when lps_dataBrk flag is TRUE; otherwise this field is undefined. The flag lps_vcidChgFlag, when TRUE, specifies a change of VCID with the current VCDU. The flag lps_dataBrk, when TRUE, specifies either a change of VCID with the current VCDU or unknown number of missing VCDUs just prior to the current VCDU. lps_pcdCnt is the number of lps_vcduInfo values actually written by MFPS to the shared memory block. lps_subIntvId provides the Id of the sub interval to which the VCDU belongs. PCDS uses lps_subIntvId to retrieve more information about the sub interval from the database. lps_eocFlag, when TRUE, specifies that the End. of Contact has been reached within this VCDU set.

5.2.4 IPC Mechanism(s)

The mechanisms for PCDS to receive data from MFPS uses shared memory and database tables. Shared memory will be arranged so that each block of memory contains one lps_mainPcdInfoStruct structure. Access to the shared memory is done using global shared memory access routines. The project database is used for the storage of information about each subinterval using the sub interval Id as the key. This was chosen in order to allow convenient access by all subsystems, including PCDS, to the subinterval information.

5.3 With LPS Data Transfer Subsystem

5.3.1 Purpose of Interface

This interface provides LDTS with the calibration file and the MSCD file from MFPS on a subinterval basis.

5.3.2 Description of Interaction

MFPS writes the calibration data file (Cal_File) and the mirror scan correction data file (MSCD_File) to a disk on a subinterval basis. MFPS then updates the LPS_File_Info information in the LPS database on a subinterval basis which includes the names of the Cal_File and MSCD_File for access by all subsystems using sub-interval Id as the key.

The Cal_File contains the deinterleaved and aligned calibration data according to the bandwidth and the detectors for a subinterval (Sub_Intv), the major frame time, and the status data from every major frame. The MSCD_File contains the second half scan error (SHS_Err), first half scan error (FHS_Err), and the scan direction (Scan_Dir) associated with a particular subinterval. Additionally, the MSCD_File contains the major frame time for every major frame.

MFPS updates the Sub_Intv information in the LPS database on a subinterval basis. The Sub_Intv includes the contact id (Contact_Sequence_Id), the file version number (File_Version_No), the subinterval id (Sub_Intv_Sequence_Id), the subinterval start time (MF_Start_Time), the subinterval stop time (MF_Stop_Time), and the virtual channel id (VCID).

5.3.3 Dataflow Structure

The format of the Cal_File and the MSCD_File are defined in the LPS Data Format Control Book.

5.3.4 IPC Mechanism(s)

The Unix files on a shared system disk will be used as the IPC mechanism between the MFPS and the LDTS. The LDTS uses sub interval Id as the key to get the names of CAL and Mscd files from the LPS database. It then can access the files on the shared disk. The project database is used for getting sub interval Ids for a contact period. This was chosen in order to allow convenient access by all subsystems, including LDTS, to

the subinterval information.

5.4 With Management And Control Subsystem

5.4.1 Purpose of Interface

This interface allows MACS to obtain the MFPS QA information (MFP_Acct) and subinterval information (Sub_Intv) from MFPS for the purpose of creating LPS QA reports. MACS also uses this interface to get information about calibration data file (Cal_File) and the mirror scan correction data file (MSCD_File) on a sub-interval basis for creating meta data files.

5.4.2 Description of Interaction

MFPS generates the quality and accounting information (MFP_Acct) and subinterval information (Sub_Intv) on a subinterval basis and stores it into the LPS database from where it is available to all subsystems, including MACS. MFPS also creates Cal_File and MSCD_File on a sub-interval basis and stores information about these files (LPS_File_Info) into the database.

5.4.3 Dataflow Structure

The following shows the contents of MFPS quality and accounting information (MFP_Acct) schema which is updated by the MFPS:

Data Structure	Type	Bytes
MFP_Acct	struct	84+
Sub_Intv_Sequence_Id	int	4
Mjf_CADU_Rcvd_Cnt	int	4
Mjf_CADU_Fly_Cnt	int	4
Mjf_CADU_Polarity_Cnt	int	4
Mjf_CADU_Polarity_Chg_Cnt	int	4
Mjf_CADU_Bit_Slip_Cnt	int	4

Mjf_CADU_Sync_Err_Cnt	int	4
Mjf_CADU_Missing_Cnt	int	4
Mjf_CADU_RS_Corr_Cnt	int	4
Mjf_CADU_RS_Uncorr_Cnt	int	4
Mjf_CADU_BCH_Data_Corr_Cnt	int	4
Mjf_CADU_BCH_Data_Uncorr_Cnt	int	4
Mjf_CADU_BCH_Data_Bits_Corr	int	4
Mjf_CADU_BCH_Pointer_Corr_Cnt	int	4
Mjf_CADU_BCH_Pointer_Uncorr_Cnt	int	4
Mjf_CADU_BCH_Pointer_Bits_Corr	int	4
Mjf_CADU_CRC_Err_Cnt	int	4
Mjf_CADU_Seq_Err_Cnt	int	4
Mjf_Cnt	int	4
Mjf_Tossed_Cnt	int	4
Mjf_Eol_Err_Cnt	int	4
Mnf_Ctr_Err_Cnt	int	4
Mjf_Time_Code_Err_Cnt	int	4
Mjf_Full_Fill_Cnt	int	4
Mjf_Part_Fill_Cnt	int	4

The following structure shows the contents of the subinterval information (Sub_Intv) schema which is updated by the MFPS:

Data Structure	**Type**	**Bytes**
Sub_Intv	struct	63
Contact_Sequence_Id	int	4
File_Version_Num	int	4
Sub_Intv_Sequence_Id	int	4
MF_Start_Time	char	25
MF_Stop_Time	char	25
VCID	byte	1

The following structure shows the contents of the LPS_File_info

schema which is updated by the MFPS:

Data Structure	Type	Bytes
LPS_File_Info	struct	529
Sub_Intv_Sequence_Id	int	4
File_Path	char	256
File_Name	char	256
File_Type	char	13

5.4.4 Mechanism(s)

The IPC mechanism between the MFPS and the MACS involves the LPS database. The database allows convenient access by MACS to the subinterval information (Sub_Intv), the QA information (MFP_Acct), and the names of Cal_File and MSCD_File which are stored in the database by the MFPS.

The MACS, using contact Id, retrieves the sub interval Id(s) from the database and using the sub interval Ids, it is able to retrieve QA information for generating the LPS reports for printer and terminal,and for getting the names of Cal_File and MSCD_File for generating the meta data.

6 Payload Correction Data Processing Subsystem

6.1 With Image Data Processing Subsystem

6.1.1 Purpose of Interface

The interface between the Image Data Processing Subsystem (IDPS) and the PCDS allows the IDPS to receive the WRS scene center identification (Scene_Info) for each identified scene. The Scene_Info includes WRS path number (WRS_Path_Nominal), WRS row number (WRS_Row_Nominal), Sun elevation (Sun_Elevation), and the Scene center time (Scene_Center_Time) which accurately describes each scene. In addition, an end of contact period indicator (PCD_EOC_Flag) is sent to notify the IDPS when the contact period has ended.

6.1.2 Description of Interaction

The PCDS opens an existing FIFO during the subsystem initialization. When a WRS scene center is identified, the PCDS formats the scene center information and uses the global routine, lps_FIFOSend, to write the scene center information to the IDPS. The IDPS must retrieve the messages from the FIFO until an end of contact period indicator has been received from the PCDS.

6.1.3 Dataflow Structure

The following shows the dataflow structures used to pass the information to the IDPS subsystem.

SRS Data Structure	DDS Data Structure	Type	Bytes
Scene_Info	lpsSceneDescrStruct	struct	22
Scene_Id			
WRS_Path_Nominal	WRS_Path_Nominal	lpsUInt8Type	1
WRS_Row_Nominal	WRS_Row_Nominal	lpsUInt8Type	1
Scene_Center_Time	Scene_Center_Time	lpsTimeStruct	15

Sun_Elevation	Sun_Elevation	lpsReal32Type	4
End_Of_Contact_Flag	End_Of_Contact_Flag	lpsBooleanType	1

6.1.4 IPC Mechanism(s)

A FIFO is the chosen IPC mechanism for data transfer between the PCDS and the IDPS. A global routine, lps_FIFOSend, will be used to write the Scene_Info to the IDPS. FIFOs provide access synchronization and is an ideal mechanism for passing small structures between two processes.

6.2 With Management And Control Subsystem

6.2.1 Purpose of Interface

The interface between the PCDS and the MACS will allow the MACS to receive the PCD processing status (PCD_Assemble_Cycle_Status), and the PCD accounting (PCD_Acct). In addition, the PCDS provides the file name for each PCD file created on a subinterval basis.

6.2.2 Description of Interaction

The PCDS retrieves the PCD parameters and thresholds (PCD_Parms) from the LPS database store for setting up the PCD processing. The PCD parameters and thresholds are validated using domain constraint checking provided by the LPS database

The PCD processing status (PCD_Assemble_Cycle_Status) is sent to the LPS Journal using the global routine, lps_LogMessage, during each contact period.

After a subinterval has been processed, the PCDS generates the PCD quality and accounting information (PCD_Acct) and stores the subinterval accounting information in the PCD_Acct, PCD_Scene_Acct, and the LPS_Output_Files tables of the LPS database.

During scene identification, the PCDS generates scene center descriptions (PCD_Scene_Acct) and stores the scene descriptions into the database table, PCD_Scene_Acct, for

later metadata file generation by the MACS.

In addition to the scene descriptions, the PCDS extracts the ETM+ bands present within a subinterval. When a subinterval is determined, the bands present are extracted from the PCD major frame and stored into the database table, Bands_Present, for later metadata file generation by the MACS.

6.2.3 Dataflow Structure

The following shows the dataflow structure of the PCDS quality and accounting information (PCD_Acct).

SRS Data Structure	DDS Data Structure	Type	Bytes
PCD_Acct	pcdAccountingStruct	struct	65
{Sub_Intv_Id	Sub_Intv_Sequence_Id	lpsUInt32Type	4
PCD_File_Info			
Sub_Intv_Id			
PCD_File_Name			
Num_PCD_MJF	Num_PCD_MJF	lpsUInt16Type	2
First_PCD_MJF_Time	First_PCD_MJF_Time	lpsTimeStruct	15
Orbit_Num	Orbit_Num	lpsUInt32Type	4
4{Minor_Frame_Acct			
Num_PCD_MNF_Sync_Errors	Num_PCD_MNF_Sync_Errors	lpsUInt32Type	4
Num_PCD_Filled_MNF	Num_PCD_Filled_MNF	lpsUInt32Type	4
Failed_PCD_Votes	Failed_PCD_Votes	lpsUInt32Type	4
Major_Frame_Acct			
Num_PCD_Filled_MJF	Num_PCD_Filled_MJF	lpsUInt32Type	4
Num_Avail_ADP	Num_Avail_ADP	lpsUInt32Type	4

Num_Rejected_ADP	Num_Rejected_ADP	lpsUInt32Type	4
Num_Missing_ADP	Num_Missing_ADP	lpsUInt32Type	4
Num_Avail_EDP	Num_Avail_EDP	lpsUInt32Type	4
Num_Rejected_EDP	Num_Rejected_EDP	lpsUInt32Type	4
Num_Missing_EDP}4	Num_Missing_EDP	lpsUInt32Type	4
WRS_Path_Nominal			
WRS_Row_Nominal			
Scene_Center_Time			
Horizontal_Display_Shift			
Sun_Azimuth			
Sun_Elevation			
Cal_Door_Activity_Status}			

The following shows the dataflow structure of the PCDS scene descriptions (PCD_Scene_Acct).

SRS Data Structure	**DDS Data Structure**	**Type**	**Bytes**
PCD_Scene_Acct	lpsSceneDescrStruct	struct	TBD
{WRS_Path_Nominal	WRS_Path_Nominal	lpsUInt8Type	2
WRS_Row_Nominal	WRS_Row_Nominal	lpsUInt8Type	2
Scene_Center_Time	Scene_Center_Time	lpsTimeStruct	15
Horizontal_Display_Shift	Horizontal_Display_Shift	lpsReal32Type	4
Sun_Azimuth	Sun_Azimuth	lpsReal32Type	4
Sun_Elevation	Sun_Elevation	lpsReal32Type	4
Cal_Door_Activity_Status}	Cal_Door_Activity_Status	lpsUInt8Type	1

The following shows the dataflow structure of the bands present metadata (Bands_Present).

SRS Data Structure	DDS Data Structure	Type	Bytes
Bands_Present	lpsBandInfoStruct	struct	16
{Contact_Id			
File_Version_Number			
PCD_Cycle_Time	PCD_Cycle_Time	lpsTimeStruct	15
Band_Present	Band_Present	lpsUCharType	1

6.2.4 IPC Mechanism(s)

The LPS Journal is the chosen IPC mechanism for status message transfer between the PCDS and the MACS. The global routine, lps_LogMessage provides access synchronization for logging PCDS processing status and error messages .

The LPS database is the chosen mechanism for the PCD_Acct, PCD_Scene_Acct, and Bands_Present data transfer between the PCDS and the MACS. The LPS database is also used to transfer the PCDS file name to the MACS.

6.3 With LPS Data Transfer Subsystem

6.3.1 Purpose of Interface

The interface between the PCDS and the LDTS will provide the LDTS with the L0R file names and locations generated during the contact period processing.

6.3.2 Description of Interaction

The PCDS generates PCD files on a subinterval basis. When the processing for the contact period is complete, the generated file names and locations are passed to a global library routine called lps_db_InsertFileInfo where they are made available to

the LDTS.

6.3.3 Dataflow Structure

The dataflow structure for the interface between PCDS and LDTS is handled entirely within the global routine lps_db_InsertFileInfo.

6.3.4 IPC Mechanism(s)

The LPS database is used to transfer the PCDS file name and location to the LDTS.

7 Image Data Processing Subsystem

7.1 With LPS Data Transfer Subsystem

7.1.1 Purpose of Interface

This interface allows the LDTS subsystem to receive the Band_File and the Browse_File from the IDPS subsystem.

7.1.2 Description of Interaction

IDPS creates the Browse_File and Band_File and places them on the shared system disk.

7.1.3 Dataflow Structure

The format of the Browse_File and Band_File will be defined in the LPS - LP DAAC ICD. The Band_File consists of 3 or 6 files, one for each band. The Browse_File consists of both the multiband browse file (idp_BrowseMulti) and the browse overlay file (idp_Browse- Overlay).

7.1.4 IPC Mechanism(s)

The IPC mechanism between IDPS and LDTS will be system software performing the LDTS function. At the LDTS level, the Browse_File and Band_File will be treated as Unix files on a shared system disk.

7.2 With Management And Control Subsystem

7.2.1 Purpose of Interface

IDPS generates status messages (IDP_Status) that are passed to the MACS using the global routine lps_LogMessage. The IDPS accounting information (IDP_Acct) is stored in the database for access by MACS. The Band and Browse file names (Band_File_Names and Browse_File_Names) are stored in the database for access by MACS and LDTS. The file names are placed in the LPS_Output_Files table using a global routine.

7.2.2 Description of Interaction

After IDPS is invoked by MACS (with the parameter of lps_ContactSequenceId), IDPS in turn invokes the three child processes idp_Band, idp_Browse and idp_ACCA. Each of these child processes returns status messages to MACS via lps_LogMessage. In addition, each of the child processes produces accounting information (IDP_Acct) which is stored in the database for access by MACS. The file names (idp_browseFileNames and idp_bandFileNames) are placed in the database (LPS_Output_Files) using a global routine.

7.2.3 Dataflow Structure

The following shows the contents of IDPS quality and accounting information (IDP_Acct).

Data Structure		Type	Bytes
IDP_Acct	IDP_Acct	struct	
Sub_Intv_Id	idp_bandSubIntvIdArg	short	2
CCA_Method	idp_accaMethod	string	2
	{idp_bandSubIntvSceneNumArg	short	2
	idp_bandSceneCtrScanNumArg	short	2
{Band_Gains	Band_Gains	Boolean	1
Gain_Change_Flag	Gain_Change_Flag	Boolean	1
{ACCCA	{idp_accaSceneScores	struct	44
Scene_Id	Scene_Id	struct	14
WRS_Path_Nominal	WRS_Path_Nominal	short	2
Sun_Elevation	Sun_Elevation	float	4
Scene_Center_Time	Scene_Center_Time	double	8
CCA_Quadrant1_Score	CCA_Quadrant1_Score	double	8
CCA_Quadrant2_Score	CCA_Quadrant2_Score	double	8
CCA_Quadrant3_Score	CCA_Quadrant3_Score	double	8
CCA_Quadrant4_Score	CCA_Quadrant4_Score	double	8
CCA_Aggregate_Score	CCA_Aggregate_Score	double	8
}	}		

}	}		

The following shows the partial contents of the LPS_Output_Files table (idp_browseFileNames and idp_bandFileNames).

Data Structure		Type	Bytes
Sub_Intv_Id	idp_bandSubIntvId	short	2
Browse_File_Names	idp_browseFileNames		
Multi_Browse_File_Name	Multi_Browse_File_Name	char	50
	Overlay_File_Name	char	50
Band_File_Names	idp_bandFileNames		
Band_File_Names_Fmt1	Band_File_Names_Fmt1		
6{Band_File_Name}6	6{Band_File_Name}6	char	300
Band_File_Names_Fmt2	Band_File_Names_Fmt2		
3{Band_File_Name}3	3{Band_File_Name}3	char	150

7.2.4 IPC Mechanism(s)

The status messages to the MACS will be sent via the global routine lps_LogMessage. The Band and Browse file names will be stored in the database (LPS_Output_Files) using a global routine. Both MACS and LDTS will have access to these file names. The accounting information will be stored in the database (IDP_Acct) to which MACS will have access.

8 LPS Data Transfer Subsystem

8.1 With Management And Control Subsystem

8.1.1 Purpose of Interface

The purpose of the LDTS to MACS interface is for the logging of status and error messages generated during LDTS processing and for the sending of the File Transfer Summary report, generated by LDTS, to the MACS.

8.1.2 Description of Interaction

The LDTS function which generates the file transfer summary report is invoked by the MACS. This function, after performing the database queries and system calls necessary to generate the report contents, will send the data to the MACS for display or printing via host system services.

LDTS will invoke the global LPS function lps_LogMessage to write all status and error messages to the LPS journal.

8.1.3 Dataflow Structure

See the LPS design for the contents of the file transfer summary report.

8.1.4 IPC Mechanism(s)

For status and error messages the global routine lps_LogMessage will be used as the IPC mechanism.

File summary report data will be displayed to the operator's console as a direct result of MACS invoking the LDTS report generation routine.